

IN THE CLAIMS:

Please amend the claims as follows:

1 1. (Twice Amended) A program converting unit for generating a machine language  
2 instruction from a source program for [an embedded microprocessor series that manages an N-bit  
3 address while processing M-bit data,] a microprocessor having an address width N and a data  
4 width M, N being greater than M, N and M being [customized] inputs to the program converting  
5 unit as specified by a user, the value of N depending on a program size of the source program,  
6 said program converting unit comprising:

7 parameter holding means for holding a data width M and a pointer width N, said data  
8 width M representing the number of bits of data used in the source program, said pointer width  
9 N representing the number of bits of an address, said N and M being [designated] input by a  
10 user during an execution of the program converting unit, the value of N depending on the  
11 program size; and

12 generating means for generating an instruction to manage said data width M when a  
13 variable operated by said instruction represents the data, and for generating an instruction to  
14 manage said pointer width N when a variable operated by said instruction represents the address,

15 wherein the program converting unit generates a unique set of machine language  
16 instructions from the source program for each N specified by the user.

1 6. (Amended) A program converting unit for generating a machine language  
2 instruction based on a source program for a processor that manages an N-bit address while  
3 processing M-bit data, N being greater than M, said program converting unit comprising:

4 syntax analyzing means for analyzing a syntax of the source program to convert the same  
5 into an intermediary language comprising intermediary instructions, and subsequently for judging  
6 whether or not each variable contained in said intermediary instructions represents data used in  
7 an address;

136

8 table generating means for generating a table for each variable in said intermediary  
9 instructions, said table holding a name together with a type of each variable, said type  
10 representing one of the data and the address;

11 parameter holding means for holding a data width and a pointer width, said pointer width  
12 designated by a user as an input during an execution of the program converting unit, said data  
13 width representing the number of bits of the data while said pointer width [representing]  
14 represents the number of bits of the address; and

15 generating means for generating an instruction to manage said data width when the  
16 variable in said intermediary instruction represents the data, and an instruction to manage said  
17 pointer width when said variable represents the address.

1 13. (Twice Amended) A program converting unit for generating a machine language  
2 instruction from a source program for an embedded microprocessor series that manages an N-bit  
3 address while processing M-bit data, N being greater than M, N being [customized] an input to  
4 the program converting unit depending on a program size, said program converting unit  
5 comprising:

6 parameter holding means for holding a data width M and a pointer width N, said data  
7 width M representing the number of bits of data used in the source program, said pointer width  
8 N representing the number of bits of an address, said N being [designated] input by a user  
9 during an execution of the program converting unit, the value of N depending on the program  
10 size;

11 generating means for generating an instruction to manage said data width M when a  
12 variable operated by said instruction represents the data, and for generating an instruction to  
13 manage said pointer width N when a variable operated by said instruction represents the address;

14 option directing means for holding a user's direction for an overflow compensation, an  
15 overflow being possibly caused by an arithmetic operation; and

16 compensate instruction generating means for generating a compensation instruction to  
17 compensate an overflow in accordance with a type of a variable used in the arithmetic operation,  
18 said compensation instruction being generated when an effective bit-width of a variable  
19 designated by an operand is shorter than a register of N-bit wide and the arithmetic operation  
20 instruction will possibly cause an overflow exceeding said effective bit-width; and

21 prohibition means for prohibiting a generation of a compensation instruction by the  
22 compensate instruction generating means when the option directing means is storing an indication  
23 denoting not to compensate, wherein the program converting unit converts the source program  
24 into one of a plurality of different machine language programs depending on the values of N and  
25 M.

1 20. (Twice Amended) A program converting unit for generating a machine language  
2 instruction based on a source program for a processor that manages an N-bit address while  
3 processing M-bit data, N being greater than M, said program converting unit comprising:

4 syntax analyzing means for analyzing a syntax of the source program to convert the same  
5 into an intermediary language comprising intermediate instructions, and subsequently for judging  
6 whether or not each variable contained in said intermediary instructions represents data used in  
7 an address;

8 table generating means for generating a table for each variable in said intermediary  
9 instructions, said table holding a name together with a type of each variable, said type repre-  
10 senting one of the data and the address, and one of signed and unsigned data;

11 parameter holding means for holding a data width and a pointer width designated by a  
12 user during an execution of the program converting unit, said data width representing the  
13 number of bits of the data, said pointer width representing the number of bits of the address;

14 option directing means for holding a user's direction for an overflow compensation, an  
15 overflow being possibly caused by an arithmetic operation;

16 generating means for generating an instruction to manage said data width when the  
17 variable in said intermediary instruction represents the data, and an instruction to manage said  
18 pointer width when said variable represents the address; and

19 compensate instruction generating means for generating a compensation instruction to  
20 compensate an overflow in accordance with a type of a variable used in the arithmetic operation,  
21 said type being judged when said option directing means holds the user's direction for executing  
22 the overflow compensation, said compensation instruction being generated when an effective bit-  
23 width of a variable designated by an operand is shorter than a register of N-bit wide and the  
24 arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-  
25 width; and

26 prohibition means for prohibiting a generation of a compensation instruction by the  
27 compensate instruction generating means when the option directing means is storing an indication  
28 denoting not to compensate.

1 27. (Twice Amended) A program converting unit for generating a machine language  
2 instruction based on a source program for a processor that manages an N-bit address while  
3 processing M-bit data, N being greater than M, said program converting unit comprising:

4 syntax analyzing means for analyzing a syntax of the source program to convert the same  
5 into an intermediary language comprising intermediary instructions, and subsequently for judging  
6 whether or not each variable contained in said intermediary instructions represents data used in  
7 an address;

8 table generating means for generating a table for each variable in said intermediary  
9 instructions, said table holding a name together with a type of each variable, said type repre-  
10 senting one of the data and the address, and one of signed and unsigned data;

11 parameter holding means for holding a data width and a pointer width designated by a  
12 user during an execution of the program converting unit, said data width representing the  
13 number of bits of the data, said pointer width representing the number of bits of the address;

option directing means for holding a user's direction for an overflow compensation, an overflow being possibly caused by an arithmetic operation;

generating means for generating an instruction to manage said data width when the variable in said intermediary instruction represents the data, and an instruction to manage said pointer width when said variable represents the address;

compensate instruction generating means for generating a compensation instruction to compensate an overflow in accordance with a type of a variable used in the arithmetic operation, said type being judged when said option directing means holds the user's direction for executing the overflow compensation, said compensation instruction being generated when an effective bit-width of a variable designated by an operand is shorter than a register of N-bit wide and the arithmetic operation instruction will possibly cause an overflow exceeding said effective bit-width; and

prohibition means for prohibiting a generation of a compensation instruction by the compensate instruction generating means when the option directing means is storing an indication denoting not to compensate, wherein said generating means includes:

determining means for determining a kind of the machine language instruction, the machine language instruction including (1) an instruction to access to a memory, (2) an instruction to use a register, and (3) an instruction to use an immediate;

memory managing means for outputting a direction, in case of the (1) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective memory access width depending on the type of a variable to be accessed shown in said table;

register managing means for outputting a direction, in case of the (2) instruction, to manage a corresponding bit-width held in said parameter holding means as an effective bit-width depending on the type of a variable to be read/written from/in the register shown in said table;

immediate managing means for outputting a direction, in case of the (3) instruction, to manage a corresponding bit-width held in said parameter holding means for the immediate as an effective bit-width depending on the type of the immediate shown in said table; and

41 code generating means for generating the machine language instruction in accordance  
42 with the directions from said memory managing means, said register managing means, and said  
43 immediate managing means, and wherein

44 said compensate instruction generating means includes:

45 instruction judging means for judging an arithmetic operation instruction that will possibly  
46 cause an overflow for all the machine language instructions when said option instructing means  
47 holds the user's direction for executing the overflow compensation;

48 determining [judging] means, with respect to a variable in the arithmetic operation  
49 instructions determined by said instruction determining means, for determining an effective bit-  
50 width and whether said variable is signed or unsigned by referring to said table;

51 sign-extension instruction generating means for generating a compensation instruction in  
52 case of a signed variable, a logical value of a sign bit being filled into all bits higher than the  
53 effective bit-width in a register that is to store said signed variable by said sign-extension  
54 compensation instruction; and

55 zero-extension instruction generating means for generating a zero-extension compensation  
56 instruction in case of an unsigned variable, a logical value "0" being filled into all bits higher  
57 than the effective bit width in a register that is to store said unsigned variable by said zero-  
58 extension compensation instruction.

1 28. Twice Amended) A processor being one out of an embedded custom processor  
2 series of processors with different address bit widths, having an address bit width which can be  
3 [customized] selected by a user in accordance with a program size, comprising:  
4 memory means for storing a program, the memory means having a minimum storage  
5 capacity of  $2^N$  bytes to store the program and having N addresss lines, the program including  
6 an N-bit data arithmetic operation instruction and other instructions operating on both N-bit and  
7 M-bit data, N being greater than M; and

8 a processor core having an address bus of N bits which is equal in size to the number of  
9 address lines of the memory means, the processor core being selected from a plurality of  
10 processor cores and comprising:

11 a program counter for holding an N-bit instruction address to output [the same] an  
12 instruction at the N-bit address to the [said] memory means, the program counter having an N-  
13 bit address length which is equal in size to the number of address lines of the memory means;

14 fetching means for fetching an instruction from [said] the memory means using the N-bit  
15 instruction address from said program counter; and

16 executing means for executing all arithmetic operation instructions at N-bit length and  
17 for executing other instructions [aside from] except for arithmetic operation instructions at one  
18 of N-bit length and M-bit length, the executing means having N-bit length;

19 [wherein said memory means has a storage capacity equivalent of up to  $2^N$  bytes, and]  
20 whereby an N-bit address is calculated by the N-bit arithmetic operation independently of a data  
21 bit-width, said data bit-width being M.

35  
44. (Twice Amended) A processor for operating certain data in accordance with an  
1 instruction in a program, comprising:  
2  
3 a first register means for holding N-bit data;  
4 a second register means for holding N-bit data;  
5 sign-extending means for extending said M-bit data to N bits by copying an MSB of said  
6 M-bit data in a direction of an upper order, M being less than N;  
7 zero-extending means for extending said M-bit data to N bits by copying a value "0" in  
8 a direction of an upper order;  
9 operating means for operating an arithmetic operation in accordance with an instruction;  
10 instruction control means for decoding an instruction to zero-extend M-bit immediate data  
11 when said M-bit immediate data are to be stored in said first register means by the decoded  
12 instruction and to sign-extend said M-bit immediate data when said M-bit immediate data are to

13 be stored in said second register means by the decoded instruction, said zero-extended and sign-  
14 extended N-bit immediate data being outputted in one of two methods, one method being to send  
15 the extended N-bit immediate data from their respective extending means to their respective  
16 register means directly, the other being to send the same via the operating means to their  
17 respective register means, with said instruction including an indication for storing in the first  
18 register means and said instruction including an indication for storing in the second register  
19 means being of two different kinds of [instruction] instructions, both kinds of instructions having  
20 a same operation code but having different destination operands.

Claim 47 (Twice Amended), line 23, delete "both".

Claim 54 (Twice Amended), line 6, delete "customized" and insert --input--.